

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**MODELOS DE TEST ADAPTATIVOS PARA EVALUACIÓN
AUTOMÁTICA**

Sarah Dobber
Tutor: Sacha Gómez Moñivas
Ponente: Simone Santini

MAYO 2017

Modelos de test Adaptativos para la Evaluación Automática

AUTOR: Sarah Dobber
TUTOR: Sacha Gómez Moñivas

Escuela Politécnica Superior
Universidad Autónoma de Madrid
MAYO de 2017

Resumen

La introducción de la tecnología en los procesos de enseñanza y, sobre todo, en el aprendizaje tiene muchas posibilidades prometedoras. Este trabajo se centra en la ayuda que estas tecnologías pueden aportar en el proceso de aprendizaje del estudiante y en su evaluación. *Computerized Adaptive Testing* (CAT) es un tipo de pruebas realizadas a ordenador en las cuales los ítems siguientes de la misma son seleccionados en función de las respuestas proporcionadas por el examinado a ítems anteriores. Para crear exámenes con CAT es necesaria la implementación de modelos o algoritmos que tengan la función de adaptar el examen al nivel de competencia del examinado y estimar dicho nivel de competencia. Los modelos deben elegir la siguiente pregunta que se le presentará al examinado en base a las respondidas anteriormente.

Basándose en CAT, en este Trabajo de Fin de Grado se presenta un sistema de evaluación de modelos adaptativos de test. El sistema simula estudiantes contestando a preguntas en un examen tipo test en fase de aprendizaje y evaluación. Los modelos a evaluar deciden dinámicamente la secuencia de preguntas en base al conocimiento del estudiante.

Se han implementado tres modelos de control, de los cuales destaca el modelo directo. Este modelo se centra en la parte básica del temario. Con esto se consigue que los estudiantes se aprendan la parte básica del temario antes que la avanzada.

Este proyecto proporciona un método estructurado en el que se determina la validez de los modelos adaptativos.

Palabras clave

Computerized Adaptive Testing, Teoría de respuesta al ítem, Sistemas Adaptativos, Metodologías de Evaluación, Aprendizaje adaptativo, Aprendizaje personalizado

Abstract

The introduction of technology into the teaching-learning environment has many promising possibilities. This paper focuses on how these technologies can contribute to the student's learning process and evaluation. Computerized adaptive testing (CAT) is a form of computer-based test that adapts to the examinee's ability level. In other words, it is a form of computer-administered test in which the next item or set of items selected to be administered depends on the correctness of the test taker's responses to the most recent items administered. To create exams using CAT, it is necessary to implement models or algorithms that estimate and adapt the exam to the level of proficiency of the examinee. These models decide the next question that will be administered to the examinee based on the previous responses.

Based on CAT, this Bachelor Thesis presents a system that evaluates adaptive test models. The system simulates students answering questions in a test in the learning phase and evaluation phase. The models to be evaluated decide the sequence of the test questions dynamically based on the student's knowledge.

Three models are proposed and have been evaluated. The one that stands out is the “direct” model. This model focuses on the basic part of the subject, forcing students to learn the basics before they progress to more advanced material.

This project provides a structured method in which the validity of adaptive models is determined.

Keywords

Computerized Adaptive Testing, Item Response Theory, Adaptive systems, Evaluation methodologies, Adaptive Learning, Personalized Learning.

Agradecimientos

Gracias a Sacha por brindarme la oportunidad de realizar este TFG, confiar en mí y dejarme trabajar a mi ritmo.

Gracias a mi familia, a Rodri y a mis amigos por el apoyo y ayuda que me habéis dado.

Gracias a mis compañeros de la carrera, profesores, Diego, Lorenzo y todos los trabajadores de la EPS por hacer estos cuatro años de carrera más interesantes y llevaderos.

ÍNDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Alcance y objetivos.....	2
1.3	Organización de la memoria.....	2
2	Estado del arte.....	4
2.1	Computerized Adaptive Testing.....	4
3	Introducción de CAT e IRT.....	7
3.1	Computerized Adaptive Testing.....	7
3.1.1	División de preguntas en niveles.....	7
3.1.2	Funcionamiento.....	8
3.1.3	Ventajas de CAT.....	9
3.1.4	Desventajas de CAT.....	10
3.2	Item response theory.....	10
4	Diseño.....	12
4.1	Análisis de requisitos.....	12
4.1.1	Requisitos funcionales.....	12
4.1.2	Requisitos no funcionales.....	13
4.2	Tipos de alumnos.....	13
4.3	Diagrama de clases.....	13
5	Desarrollo.....	17
5.1	Modelos implementados.....	17
5.1.1	Introducción.....	17
5.1.2	Modelo tradicional.....	17
5.1.3	Modelo promoción-democión.....	17
5.1.4	Modelo directo.....	19
5.1.5	Modelo de nota mínima.....	20
5.2	Simulación.....	21
5.2.1	Configuración.....	22
5.2.2	Fase de autoevaluación.....	23
5.2.3	Fase de evaluación.....	24
6	Integración, pruebas y resultados.....	27
6.1	Parámetros utilizados.....	27
6.1.1	Modelo tradicional.....	27
6.1.2	Modelo promoción-democión.....	27
6.1.3	Modelo directo.....	27
6.1.4	Modelo nota mínima.....	28
6.2	Fase de autoevaluación.....	28
6.3	Fase de evaluación.....	34
7	Conclusiones y trabajo futuro.....	37
7.1	Conclusiones.....	37
7.2	Trabajo futuro.....	37
	Referencias.....	39
	Glosario.....	40
	Anexos.....	I
A	Manual de instalación.....	I
B	Manual del programador.....	III

ÍNDICE DE FIGURAS

FIGURA 3-1. DIAGRAMA DE FLUJO DE UN EXAMEN CAT PROPUESTO EN [3].....	9
FIGURA 4-1. DIAGRAMA DE CLASES.....	15
FIGURA 5-1. DIAGRAMA DE FLUJO DEL MODELO PROMOCIÓN-DEMOCIÓN.....	18
FIGURA 5-2. DIAGRAMA DE FLUJO DEL MODELO DIRECTO	20
FIGURA 5-3. CURVA GENÉRICA DE LA FUNCIÓN SIGMOIDEA	24
FIGURA 6-1. APRENDIZAJE EN EL NIVEL 1 DE TODOS LOS ESTUDIANTES POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	29
FIGURA 6-2. APRENDIZAJE EN EL NIVEL 2 DE TODOS LOS ESTUDIANTES POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	29
FIGURA 6-3. APRENDIZAJE EN EL NIVEL 3 DE TODOS LOS ESTUDIANTES POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	30
FIGURA 6-4. APRENDIZAJE EN TODOS LOS NIVELES DE LOS ESTUDIANTES POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	30
FIGURA 6-5. APRENDIZAJE DEL ESTUDIANTE DIRECTO POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	31
FIGURA 6-6. APRENDIZAJE DEL ESTUDIANTE INVERSO POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN.....	32
FIGURA 6-7. APRENDIZAJE DEL ESTUDIANTE BUENO POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN	33
FIGURA 6-8. APRENDIZAJE DEL ESTUDIANTE MALO POR MODELO DESPUÉS DE REALIZAR LA PRUEBA DE AUTOEVALUACIÓN	33
FIGURA 6-9. PORCENTAJE DE LOS ESTUDIANTES QUE APRUEBAN EN CADA MODELO	35

ÍNDICE DE TABLAS

TABLA 3-1. EJEMPLO DE PREGUNTAS DE CADA UNO DE LOS NIVELES	8
TABLA 4-1. DEFINICIÓN DEL ALGORITMO DE PROMOCIÓN-DEMOCIÓN CON LOS PARÁMETROS USADOS	13
TABLA 4-2. DESCRIPCIÓN DEL DIAGRAMA DE CLASES.....	14
TABLA 5-1. PARÁMETROS DE LA SIMULACIÓN INCLUIDOS EN EL FICHERO SIMULACIÓN.XML	23
TABLA 5-2. PARÁMETROS DE LOS MODELOS INCLUIDOS EN EL FICHERO SIMULACIÓN.XML.....	23
TABLA 5-3. PARÁMETROS DE LOS ALUMNOS INCLUIDOS EN EL FICHERO SIMULACIÓN.XML	23
TABLA 6-1. DEFINICIÓN DEL ALGORITMO DE PROMOCIÓN-DEMOCIÓN CON LOS PARÁMETROS USADOS	27
TABLA 6-2. DEFINICIÓN DEL ALGORITMO DIRECTO CON SU PARÁMETRO P.....	28
TABLA 6-3. DEFINICIÓN DEL ALGORITMO DE NOTA MÍNIMA CON SU PARÁMETRO P	28

1 Introducción

A continuación se detalla cual es la motivación de este Trabajo de Fin de Grado, el alcance del proyecto y los objetivos propuestos. También se explicará la estructura que tiene el documento.

1.1 Motivación

La revolución que ha supuesto la informática ha traído mejoras inmensas en distintas áreas. Estas mejoras han permitido al ser humano librarse de tareas repetitivas y dedicar más esfuerzo a tareas verdaderamente interesantes.

La educación se ha beneficiado de los avances de la informática, pero a un ritmo mucho más lento que otras áreas. Los exámenes tipo test tradicionales de longitud fija y realizados con papel y lápiz (P&P) son uno de los pilares de la educación desde que se desarrolló a principios del siglo XX. Aunque cada día es más habitual el aprendizaje asistido por tecnologías (TEL), con el uso de ordenadores en las aulas, Internet como recurso docente o la introducción de pizarras digitales, una gran parte de la educación ha permanecido igual, anclada en modelos tradicionales. [1] La mayoría de los exámenes administrados hoy en día se siguen basando en la misma idea de cuestionarios creados cada año y corregidos a mano por los profesores.

Este tipo de pruebas pueden ser problemáticas ya que, al crearse un nuevo conjunto de preguntas cada año, puede que alguna esté mal formulada y que no se descubra hasta la corrección del examen. Otro problema es la cantidad de tiempo que se pasan los profesores realizando tareas repetitivas al corregir los exámenes, el cual se podría evitar. Este tiempo también es valioso para el estudiante, ya que el *feedback* es una parte muy importante en la educación, sobretodo en exámenes de autoevaluación.

Para poder medir el nivel de conocimiento de la materia de todos los estudiantes, el examen debe tener ítems contenidos en un amplio rango de conocimiento, es decir, tener ítems fáciles para los estudiantes con menor nivel de conocimiento e ítems difíciles para los estudiantes con mayor nivel de conocimiento. Esto puede ser un problema, ya que los estudiantes que tienen mayor conocimiento de la materia tendrán que pasar por un número alto de preguntas demasiado fáciles antes de alcanzar a las preguntas que se ajustan a su conocimiento. Los estudiantes con menor conocimiento de la materia observarán que las preguntas no se ajustan a su nivel de conocimiento y se les presentarán preguntas demasiado difíciles que, posiblemente, les desalienten.

Computerized Adaptive Testing (CAT) es un tipo de pruebas realizadas a ordenador en las cuales los ítems siguientes de la misma son seleccionados en función de las respuestas proporcionadas por el examinado a ítems anteriores. [2]

Esto abre un abanico de posibilidades. El ordenador tiene la habilidad de proporcionar exámenes de tipo test a muchos examinandos simultáneamente, con la misma o con diferentes pruebas. También se puede permitir que cada examinado responda a las preguntas de la prueba a su propia velocidad. Así, el ordenador puede presentar a distintos estudiantes distintos exámenes adaptados a su nivel de competencia.

En este Trabajo de Fin de Grado se pretende crear una herramienta que ponga a prueba modelos de test adaptativos para la evaluación automática. También se implementarán varios modelos de test adaptativos (tanto en etapa de evaluación como de autoevaluación) como modelos de prueba de la herramienta.

1.2 Alcance y objetivos

El objetivo de este Trabajo de Fin de Grado es establecer un marco de evaluación de modelos de *Computerized Adaptive Testing* (CAT). Para ello se ha desarrollado un sistema capaz de simular gran cantidad de estudiantes, tanto en fase de aprendizaje (auto-evaluación) como en fase de examen, contestando a una serie de preguntas de examen cuyo orden es determinado por los modelos a evaluar. Como control de validez del sistema, se han evaluado varios modelos ejemplo y se incluyen los datos resultantes así como la recomendación resultante.

El marco desarrollado sirve para confirmar o desmentir suposiciones como “Es posible el uso de exámenes adaptativos para acelerar el aprendizaje del estudiante.”

Este Trabajo de Fin de Grado solo tendrá en cuenta exámenes de tipo test y se centrará en la parte técnica de CAT, no en la psicométrica.

1.3 Organización de la memoria

- Estado del arte: se presenta brevemente el modelo de desarrollo de CAT y su historia.
- Introducción de CAT e IRT: una exposición de los conceptos fundamentales de CAT que se han tenido en cuenta al desarrollar la aplicación. Se explica el funcionamiento de CAT, la división de los ítems en niveles y sus ventajas y desventajas. También se explica brevemente *Item Response Theory* (IRT).
- Diseño: se presenta un análisis de requisitos de la aplicación y el diagrama de clases. También se explica el tipo de alumnos utilizados en la misma.
- Desarrollo: se explica en profundidad los modelos de prueba implementados en la aplicación y el sistema implementado.
- Integración, pruebas y resultados: se presentan los resultados de las simulaciones para cada uno de los modelos implementados y se comparan con el modelo de test tradicional (P&P).
- Conclusiones y trabajo futuro: se presentan las conclusiones y las posibles mejoras de la aplicación en el futuro.

2 Estado del arte

Los exámenes tipo test tradicionales de longitud fija y realizados con papel y lápiz (P&P) son uno de los pilares de la educación desde que se desarrolló a principios del siglo XX. La mayoría de los exámenes administrados hoy en día son de este tipo con la excepción de los test de inteligencia y test de diagnóstico. Hasta principios de los años 70 todos los exámenes se evaluaban utilizando teoría de examinación clásica.

Un problema crítico al que se enfrenta este tipo de prueba es que debe tener probar un amplio rango de conocimiento. Para poder medir el nivel de conocimiento de la materia de todos los estudiantes, el examen debe contener ítems cuyas dificultades coincidan con este rango, es decir, tener ítems fáciles para los estudiantes con menor nivel de conocimiento e ítems difíciles para los estudiantes con mayor nivel de conocimiento.

La consecuencia de esta estructura de examen ha sido históricamente que los examinados más competentes han tenido que vadear a través de un gran número de ítems demasiado fáciles antes de llegar a un ítem que se ajusta a su nivel de conocimiento (y que proporcione información sustancial al examinador a la hora de evaluar). Los examinados menos competentes se enfrentan un problema diferente. Para ellos, los ítems fáciles se ajustan a su nivel de conocimiento, mientras que los difíciles proporcionan poca información al examinador. [3]

En los años 80 los ordenadores empezaron a afectar varios aspectos de la sociedad y, específicamente, el de los exámenes. La disponibilidad de grandes ordenadores posibilitó la implementación de *Item Response Theory* (IRT), lo que ha conducido a una investigación y desarrollo cada vez más sofisticados con el objetivo de mejorar los procedimientos de los exámenes. En los años 80, la disponibilidad de ordenadores potentes y pequeños (para la época) condujo a la investigación y el desarrollo de pruebas adaptativas: *Computerized Adaptive Testing* (CAT). [2]

2.1 Computerized Adaptive Testing

A lo largo de toda la historia siempre se ha buscado el equilibrio entre los exámenes individuales y exámenes colectivos. Los exámenes individuales tienen la ventaja de que no contienen preguntas inapropiadas y, además, asegura que el examinado comprende la tarea. Los exámenes en grupo tienen la ventaja de ser una situación uniforme para todos los examinados, así como la enorme reducción del coste de evaluación. [3]

La manera más eficaz de evaluar a un examinado es con un examen que contiene ítems que no son ni demasiado difíciles ni demasiado fáciles. Así, la función principal del computador es, en cada paso del examen, estimar el nivel de habilidad del examinado, basándose en las respuestas de ítems anteriores. [4]

Computerized Adaptive Testing (CAT) es un tipo de pruebas realizadas a ordenador en las cuales los ítems siguientes de la misma son seleccionados en función de las respuestas proporcionadas por el examinado a ítems anteriores. [2]

Desde el punto de vista de los estudiantes, el examen se adapta a su nivel de habilidad. Así, los estudiantes con mayor conocimiento no se verán “aburridos” por las preguntas fáciles, mientras que los estudiantes con menor conocimiento de la materia no se desalentarán por las preguntas difíciles del examen.

3 Introducción de CAT e IRT

3.1 Computerized Adaptive Testing

Todos los tests siguen algún tipo de **algoritmo de prueba**. Este algoritmo es un conjunto de reglas que especifican las preguntas que tiene que responder el examinado y el orden en el que éstas se le presentan. Este algoritmo se presenta en tres partes distintas:

- Como empezar: ¿Cómo elegimos el primer ítem del examen?
- Como seguir: Después de haber visto la respuesta del evaluado a la pregunta actual, ¿cómo elegimos el siguiente ítem que se va a administrar?
- Como parar: ¿Cómo sabemos cuándo parar el examen? [3]

3.1.1 División de preguntas en niveles

En exámenes convencionales donde el examen es fijo para todos los estudiantes (P&P), a todos los examinados se les presenta todos los ítems de la parte del temario evaluada. Estos ítems podrían ser seleccionados de un conjunto más grande de ítems, pero una vez elegidos se mantienen constantes a lo largo del examen. En un examen tipo CAT los examinados se presentan con exámenes individuales donde cada examen se compone de un set de ítems específico cogido de un conjunto más grande de ítems. [3]

Para aprovechar al máximo las ventajas de los exámenes de tipo CAT, el conjunto que contiene los ítems a ser seleccionados para el examen deberá contener ítems de alta calidad divididos en diferentes niveles de competencia. Aunque se puede utilizar cualquier número de niveles se han considerado tres niveles. [3]

- Nivel 1: preguntas básicas que evalúan la memorización de la materia.
- Nivel 2: preguntas que evalúan la comprensión de la materia.
- Nivel 3: preguntas avanzadas que evalúan la capacidad de aplicación de la materia.

Un ejemplo de preguntas pertenecientes a estos niveles se muestra en la tabla 3-1. Las preguntas mostradas en la tabla se corresponden con la materia de la asignatura de Programación I en el Grado de Ingeniería Informática.

Nivel 1	<p>¿Qué es un compilador?</p> <ol style="list-style-type: none">Es un programa informático que traduce un programa que ha sido escrito en un lenguaje de programación a un lenguaje de máquina.Es un programa informático capaz de analizar y ejecutar otros programas.Es un conjunto prescrito de reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos.
Nivel 2	<p>Considere el siguiente fragmento de código en un lenguaje de programación fuertemente tipificado, ¿cuál de las siguientes afirmaciones es cierta?</p>

	<pre>int a =5; String b = "hello"; a = a + b;</pre> <p>a. El compilador da un error al intentar compilar el programa. b. El programa da un error en tiempo de ejecución. c. El programa no da ningún error.</p>
Nivel 3	<p>¿Qué valor tiene la variable “x” después de la ejecución de este fragmento de código con a=11 y b=17?</p> <pre>int mifuncion (int a, int b) { if (a > b) return a + b; return mifuncion (b - a, a); }</pre> <p>a. 5 b. 28 c. No hay solución, ya que entra en un bucle infinito.</p>

Tabla 3-1. Ejemplo de preguntas de cada uno de los niveles

3.1.2 Funcionamiento

El objetivo de un examen de tipo CAT es maximizar la precisión del examen basándose en las preguntas que el estudiante ha respondido anteriormente.

Desde el punto de vista del estudiante, este es presentado con preguntas que se ajustan a su conocimiento de la materia. Por ejemplo, si el estudiante responde correctamente a las preguntas intermedias, se le presentará una pregunta más difícil mientras que si responde incorrectamente a las preguntas de nivel intermedio, se le presentará preguntas más sencillas.

El funcionamiento básico de CAT es un algoritmo iterativo con los siguientes pasos:

1. Se busca entre el conjunto de ítems el más óptimo en base a la estimación de la habilidad del estudiante.
2. Se presenta el ítem escogido al estudiante, el cual lo responderá de manera correcta o incorrecta.
3. Se actualiza la habilidad del estudiante en base a todas sus respuestas anteriores.
4. Se repiten los pasos 1-3 hasta que se cumpla el criterio de finalización del examen.

Debido a que no se conoce nada del estudiante antes del examen, la primera pregunta será de dificultad media o media-baja.

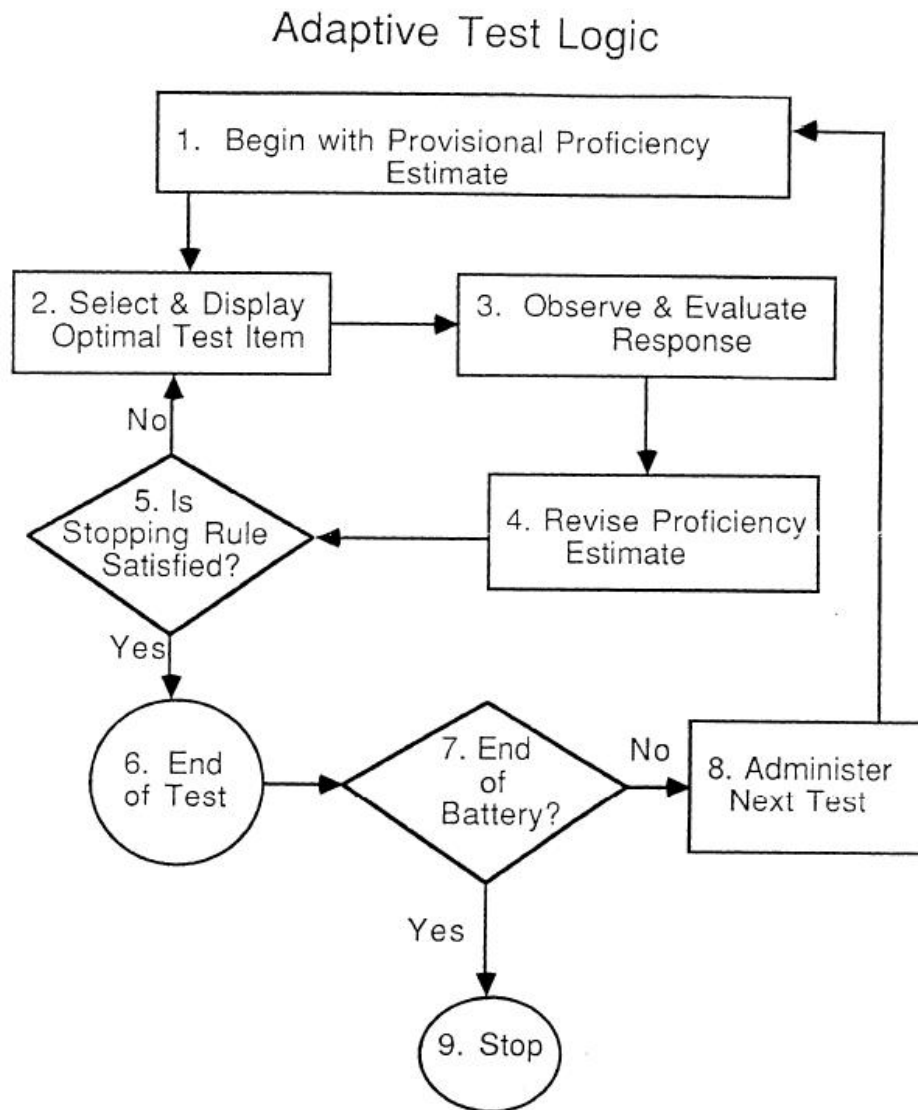


Figura 3-1. Diagrama de flujo de un examen CAT propuesto en [3]

3.1.3 Ventajas de CAT

Entre las mejoras que ofrece un examen tipo CAT frente a un examen tradicional se encuentran las siguientes: [3]

1. Hay un aumento de la seguridad de los exámenes. Esto es debido a que se almacenan el conjunto de preguntas y no solo las preguntas de un examen específico. Así, si alguien tuviera acceso a las preguntas, tendría que estudiarse un número muy alto de ellas, en cuyo caso se merecería buena nota.
2. Los examinados pueden responder a las preguntas a su ritmo y, además, el tiempo que tarda una persona en responder a una pregunta es información útil en la evaluación.

3. Los examinados invierten su tiempo productivamente ya que a todos ellos se les presentan desafíos durante el examen sin que sean demasiado difíciles para que no se desalienten.
4. Se resuelven los problemas de carácter físico de las hojas de respuestas. En este tipo de prueba no existe la ambigüedad asociada a qué respuesta está realmente marcada ni problemas asociados a la pérdida de las hojas de respuesta.
5. Los exámenes se pueden puntuar inmediatamente, dándole *feedback* inmediato al estudiante, lo cual es especialmente importante en los exámenes de autoevaluación.
6. Se pueden cambiar fácilmente los exámenes, ya que se pueden introducir nuevos ítems de manera no intrusiva.
7. Los ítems defectuosos pueden eliminarse inmediatamente.
8. Se pueden incluir varios tipos de preguntas al examen, sin la necesidad de ceñirse a las preguntas de tipo test. Se pueden incluir preguntas con ficheros de audio y vídeo, permitiendo un rango más amplio de preguntas.
9. Los exámenes de tipo CAT típicamente se pueden acortar hasta un 50% de los exámenes tradicionales manteniendo el mismo nivel de precisión que en los exámenes tradicionales. [2]

3.1.4 Desventajas de CAT

1. Los exámenes de tipo CAT requieren un gran conjunto de preguntas, lo que complica la calibración de los ítems. Si los elementos no se calibran adecuadamente, el examen no será ni válido ni fiable. [5] Parte de la calibración de los ítems consiste en la estimación de sus parámetros. Para ello necesita datos de medición reales, los cuales no siempre están disponibles para las pequeñas instituciones educativas. [6]
2. Las preguntas reflejadas en el algoritmo de prueba (necesarias para el desarrollo de un modelo CAT) presentadas en el apartado 3.1 no son fáciles de responder. Existen muchos algoritmos de comienzo, continuación y parada en CAT y la mínima variación en cualquiera de ellos genera sistemas muy distintos que tendrían efectos muy dispares en la resolución de los exámenes. Cada pequeño cambio que se haga sobre un modelo existente debe ser probado exhaustivamente para asegurar su fiabilidad y precisión. [3]

3.2 Item response theory

Item response theory (IRT) es una familia de descripciones matemáticas que describen lo que sucede cuando un examinado se encuentra con un ítem. Todos los ítems deben medir la misma competencia. IRT lo formaliza postulando explícitamente una sola dimensión de conocimiento o rasgo subyacente del cual dependen todos los ítems de la prueba, en cierta medida, de su respuesta correcta. Llamaremos a este rasgo competencia. [3] En IRT a cada examinado se le define por un valor de competencia, mientras que a cada pregunta se le asigna una dificultad. IRT se encarga de elegir la siguiente pregunta del examen en base a estos dos valores. [7]

4 Diseño

Se ha desarrollado un sistema de evaluación de modelos adaptativos de test. El sistema simula estudiantes contestando a preguntas en un examen tipo test tanto en fase de examinación como en fase de auto-evaluación y aprendizaje. El modelo a evaluar decide dinámicamente la secuencia de preguntas en base al conocimiento del estudiante.

En fase de examen, el estudiante contesta a una cantidad reducida de preguntas y la secuencia de preguntas es determinada por los modelos bajo evaluación. En fase de aprendizaje, la cantidad de preguntas es mucho más alta y el sistema simula el proceso de aprendizaje según una curva de aprendizaje sigmoidea, concretamente una integral de la distribución normal Gaussiana.

Como prueba final del sistema de evaluación de modelos, se han evaluado 3 modelos adaptativos, comparándolos con el modelo tradicional no-adaptativo, y se publican los resultados en esta memoria.

El diseño de la aplicación contiene tres distintas partes:

- Modelos implementados: se han implementado varios modelos como ejemplo. Estos modelos son el modelo de promoción-democión, que se adapta al nivel de habilidad de estudiante, ya que si responde un número P de preguntas correctamente pasa al siguiente nivel de habilidad, mientras que si responde un número de preguntas D incorrectamente baja a preguntas de menor nivel. El modelo directo se centra en los niveles básicos antes de pasar a conceptos avanzados. Finalmente, el modelo de nota mínima trata de conseguir que el alumno tenga un conocimiento mínimo en un nivel antes de pasar al siguiente nivel. Como modelo de control se ha incluido el modelo tradicional, modelo que presenta $1/N$ preguntas de cada nivel en el examen, donde N es el número de niveles existente. Todas las preguntas tienen el mismo peso.
- Simulación: esta parte es la más importante de las tres. La simulación prueba la validez de los modelos implementados y presenta los resultados en una hoja de cálculo en Excel. Los modelos se prueban con varios tipos de alumnos en la fase de aprendizaje y evaluación.
- Persistencia: los resultados de la simulación se persisten en un fichero Excel. El análisis de los datos se realizan en el mismo fichero Excel.

4.1 Análisis de requisitos

El Sistema de evaluación de modelos adaptativos de test debe cumplir los siguientes requisitos:

4.1.1 Requisitos funcionales

RF 1. Debe ser posible definir nuevos modelos en el sistema.

RF 2. El sistema debe ser capaz de ejecutar simulaciones utilizando estos modelos.

RF 3. Los modelos deben admitir parámetros configurables.

RF 4. El sistema debe simular una gran cantidad de estudiantes haciendo exámenes de evaluación.

RF 5. El sistema debe simular el aprendizaje de los estudiantes durante la fase de auto-evaluación.

RF 6. La ejecución de la simulación debe producir un archivo de datos en texto legible.

RF 7. El sistema debe generar gráficos representando los resultados de las simulaciones.

4.1.2 Requisitos no funcionales

RNF 1. El código deberá optimizarse para que la ejecución de las simulaciones tarde lo menos posible, ya que dichas simulaciones se harán con un número muy alto de estudiantes (en la simulación, 10000 estudiantes).

RNF 2. La simulación debe funcionar en un entorno Windows.

4.2 Tipos de alumnos

Las simulaciones se han realizado con cuatro tipos de alumnos. Están compuestos por tres números que representan la probabilidad de acertar una pregunta de un nivel específico, donde dicha probabilidad es un número entre 0 y 1. Así, en las simulaciones las preguntas estarán representadas por un número aleatorio y si este número es menor que el número del alumno para el mismo nivel, se considerará que el alumno ha acertado, mientras que si es mayor, se considerará que el alumno ha fallado a la pregunta.

Los cuatro tipos de alumnos utilizados en las simulaciones se muestran en la tabla 4-1. El primer alumno (A) le da más importancia al temario básico, es decir, de los primeros niveles. El segundo alumno (B) se centra en la parte avanzada del temario sin conocer la parte básica del mismo. El tercer tipo (C) se sabe el temario incluido en todos los niveles, mientras que el último tipo de alumno (D) no se sabe el temario de ningún nivel.

Tipo de alumno	Nivel		
	1	2	3
Directo (A)	0.9	0.7	0.45
Inverso (B)	0.45	0.7	0.9
Bueno (C)	0.9	0.85	0.7
Malo (D)	0.33	0.33	0.33

Tabla 4-1. Definición del algoritmo de promoción-democión con los parámetros usados

4.3 Diagrama de clases

Descripción de clases y flujo de trabajo:

1	Simulator	La clase principal (main) que ejecuta la simulación.
1.1	Persistence	Lee el archivo pasado en la línea de comandos o, por defecto, simulation.xml mediante JAXB.

1.1.1	SimulationBean ModelBean StudentBean	Persistence instancia una jerarquía de clases empezando por SimulationBean. Esta jerarquía representa las evaluaciones a ejecutar.
1.2	TestModel TestModelFactory	Para cada ModelBean configurado, Simulator llama a métodos estáticos TestModelFactory para instanciar subclases, los modelos a evaluar.
1.2.1	PromoteDemote RewardDirectStudents MinimumGrade TraditionalLM	Modelos implementados que heredan de la clase TestModel. Estos modelos tienen parámetros específicos a los cuales se dará distintos valores en la simulación.
1.3	Student	Para cada StudentBean, Simulator instancia un objeto de clase Student.
1.4	Exam Level	Simulator instancia un objeto tipo Exam para cada combinación de TestModel y Student, y ejecuta la simulación del examen n veces. Durante el examen, cada vez que el estudiante (simulado) responde a una pregunta, se invoca el TestModel para determinar el nivel (Level) de la siguiente pregunta.
1.4.1	ExamResultsBean	Esta clase se encarga de persistir los datos obtenidos de la ejecución de los exámenes en un archivo Excel.
1.5	GaussianIntegral	En fase de autoevaluación, los exámenes contienen una gran cantidad de preguntas y el estudiante aprende contestando a estas preguntas y examinando el <i>feedback</i> del sistema. El aprendizaje (grado de conocimiento de la materia) se modela en una función Sigmoidea implementada en la clase GaussianIntegral.

Tabla 4-2. Descripción del diagrama de clases

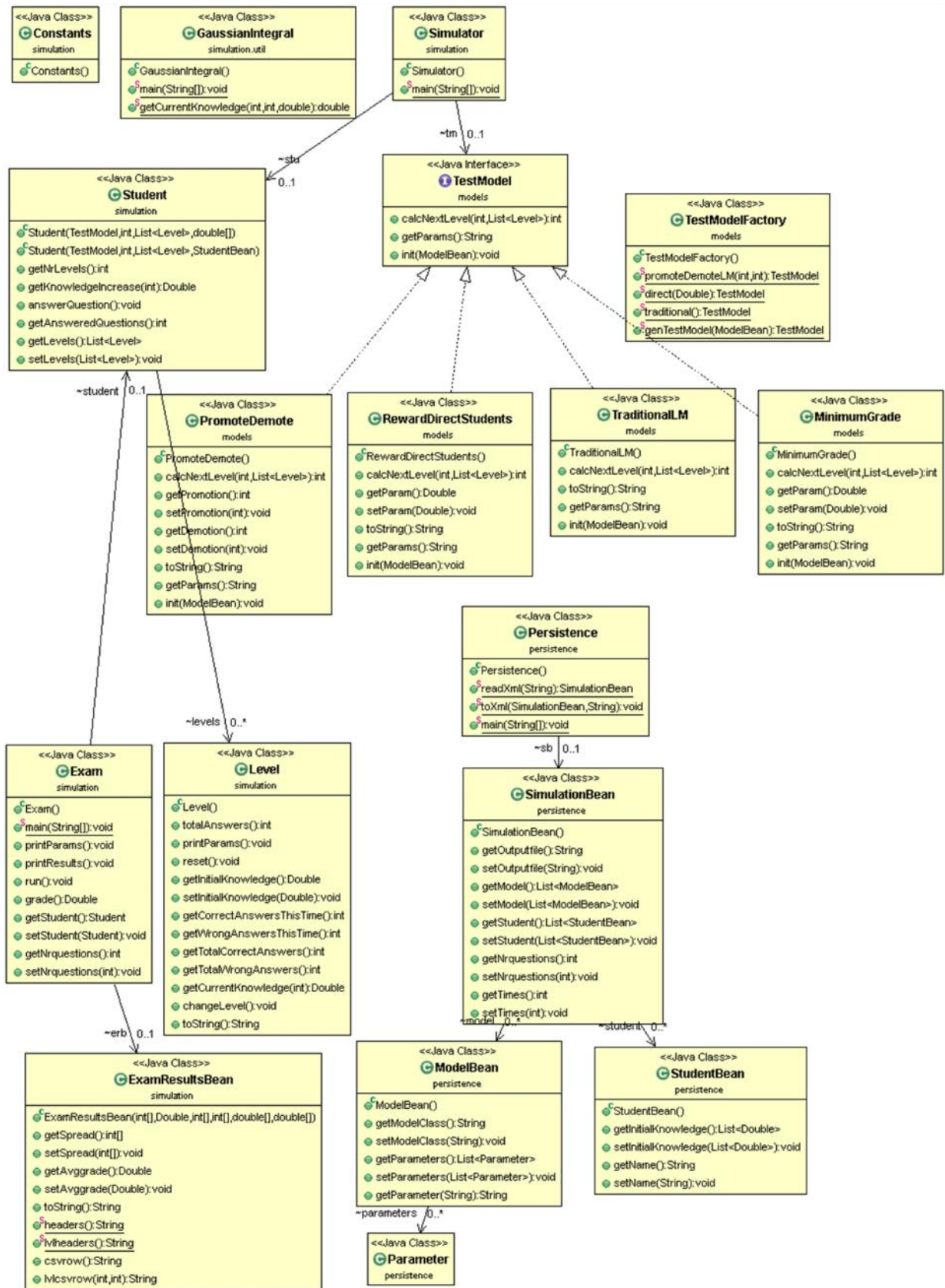


Figura 4-1. Diagrama de clases

5 Desarrollo

5.1 Modelos implementados

5.1.1 Introducción

Se han implementado tres modelos, los cuales se han metido en una simulación que se explica en el apartado 5.2. Los tres modelos siguen unas reglas básicas:

- El número de preguntas realizado en cada examen es fijo, es decir, la condición de acabado del examen es que se haber respondido un número determinado de preguntas.
- El número de niveles es fijo. (ver apartado 3.1.1)
- Las preguntas de los exámenes de los distintos alumnos varían, es decir, no todos los alumnos son presentados con el mismo examen y no siempre se le presentará con el mismo número de preguntas de cada nivel.

5.1.2 Modelo tradicional

El modelo tradicional se ha metido como modelo de control y es equivalente a una prueba realizada con papel y lápiz (P&P), donde el estudiante tiene un set de ítems fijo. El examinado será presentado con una prueba que contiene aproximadamente el mismo número de preguntas de cada nivel. Debido a que tomamos como base tres niveles de competencia distintos, las preguntas estarán categorizadas en estos niveles y el examen contendrá 1/3 de preguntas de cada nivel.

5.1.3 Modelo promoción-democión

El modelo de promoción-democión está ilustrado en la figura 5-1. En este modelo las preguntas que se le presenta al examinado serán de distintos niveles y varían entre los distintos alumnos. En este modelo el estudiante se encuentra en cada momento en un nivel determinado. Las preguntas que se presentan al estudiante en un examen dependen de las respuestas que haya dado a las anteriores. Si el estudiante responde correctamente a un número determinado de preguntas del nivel N, subirá al nivel N+1 y se le presentará una pregunta más avanzada. Sin embargo, si responde incorrectamente a un número determinado de preguntas del nivel N, bajará al nivel N-1, es decir, se le presentará una pregunta más básica.

El modelo de promoción-democión está definido por dos parámetros: el número de preguntas que hay que responder correctamente para subir de nivel y el número de preguntas que hay que responder incorrectamente para bajar de nivel. En la figura 5-1 dichas variables tienen el nombre de *nPromotion* y *nDemotion* respectivamente.

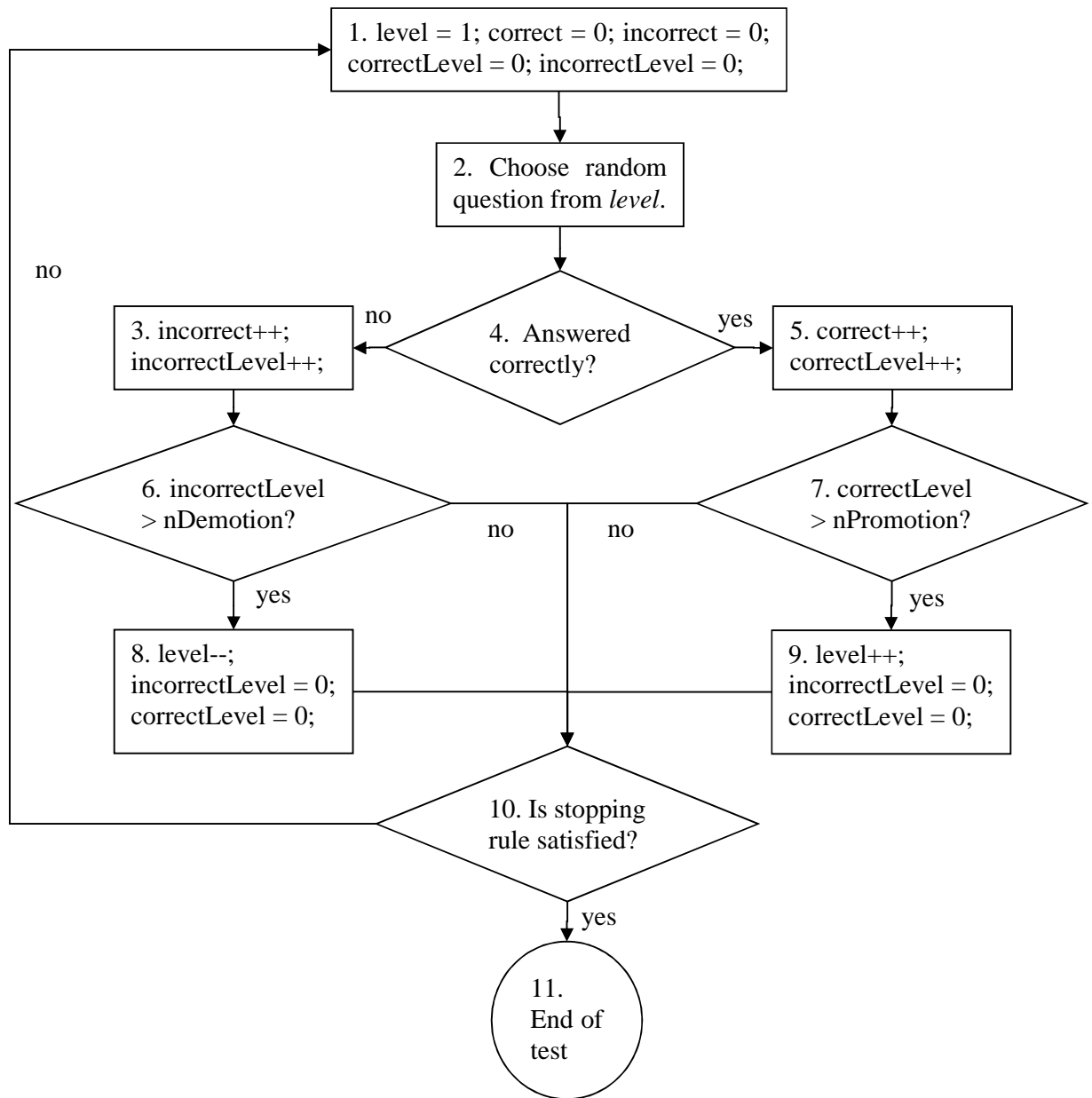


Figura 5-1. Diagrama de flujo del modelo promoción-democión

```

function recalculateCurrentLevel(current_level,
current_student) {
    if (getConsecutiveCorrectAnswers(current_level,
current_student) >= promotion) {
        promote(current_level, current_student);
    }
    if (getConsecutiveCorrectAnswers (current_level,
current_student) >= demotion) {
        demote(current_level, current_student);
    }
}

```

5.1.4 Modelo directo

El modelo directo persigue que el estudiante aprenda conceptos básicos antes de pasar a conceptos de niveles superiores, siempre intentando maximizar el resultado.

El sistema decide el nivel de la siguiente pregunta en función del conocimiento del estudiante en todos los niveles. El sistema impone un conocimiento mínimo en cada nivel como función del conocimiento del siguiente nivel. El modelo es parametrizado con la diferencia mínima necesaria entre el conocimiento a dos niveles adyacentes. Para pasar al nivel siguiente el conocimiento en el nivel actual se debe cumplir la siguiente condición:

$$k_i > k_{i+1} + p * \left(\frac{1 - k_{i+1}}{10} \right)$$

Donde

- k_i es el conocimiento del alumno en el nivel actual
- p es el parámetro

El parámetro p define el conocimiento que tienen que tener el examinado en el nivel actual en comparación con el nivel siguiente. Esto provoca que los estudiantes se centren en los niveles básicos de conocimiento antes de pasar a los niveles más avanzados. [8]

En [8] uno de los objetivos es beneficiar a los estudiantes que tienen un mayor nivel de conocimiento en los niveles básicos que en los avanzados. Por tanto, el objetivo de este modelo es crear estudiantes “directos”.

Como se puede observar en la fórmula, se calcula la diferencia entre 1 y el conocimiento en el nivel siguiente, se divide entre 10 y se multiplica por el parámetro p . Para pasar al nivel siguiente, el conocimiento en el nivel actual tiene que ser mayor que el conocimiento en el nivel siguiente más dicha diferencia.

```
function recalculateCurrentLevel(current_level,
current_student) {
    current_level = highest_level;
    iterate through levels {
        kl = knowledge at current iteration level
        knl = knowledge at next iteration level

        if (kl < knl + (1- knl)/param)
            current_level = current iteration level;
    }
}
```

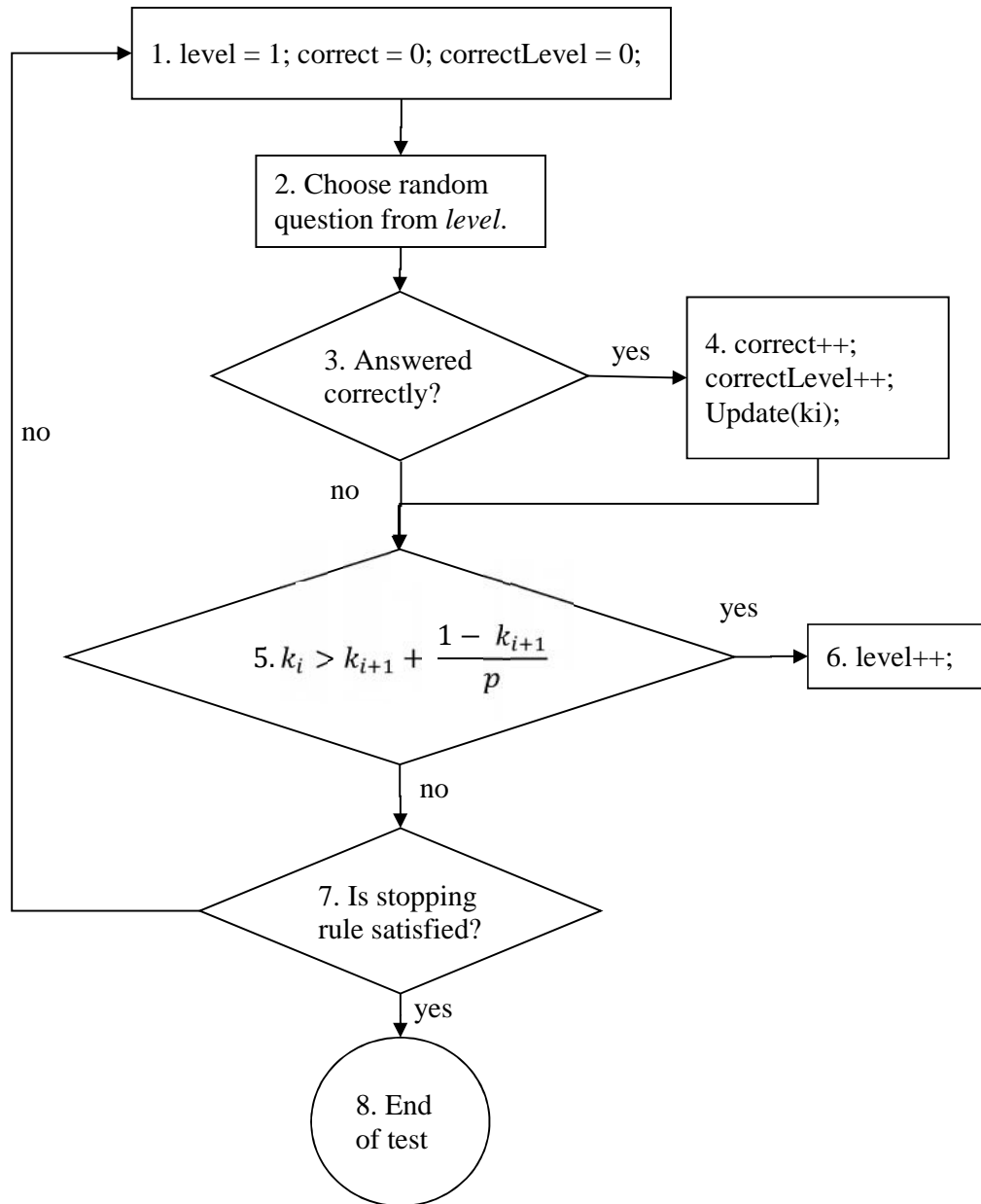


Figura 5-2. Diagrama de flujo del modelo directo

5.1.5 Modelo de nota mínima

El modelo de nota mínima trata de conseguir que el alumno tenga un conocimiento mínimo en un nivel antes de pasar al siguiente nivel. Este conocimiento mínimo es configurable. En el fragmento de código inferior se puede observar el funcionamiento del modelo. Primero se asignan las variables *mink* y *minl* que representan el mínimo conocimiento del estudiante y el mínimo nivel del estudiante. A continuación se itera sobre los niveles y se comprueba en orden ascendente si el conocimiento del estudiante en los distintos niveles es menor que *param*, que es el umbral indicado. Si es así, la función devolverá el valor de ese nivel y el estudiante deberá contestar más preguntas de ese nivel hasta que supere el umbral. Si no, se

sigue iterando hasta que eventualmente se llega al último nivel. Una vez llegado al último nivel, se devolverá el último nivel.

```
function recalculateCurrentLevel(current_level,
current_student) {
    float mink=1;
    integer minl=0;
    for (int i=0; i< levels.size(); i++) {
        k[i] = knowledge at current iteration level
        if (k[i]<param) return i;
        if (k[i]<mink) {
            mink=k[i];
            minl=i;
        }
    }
    return minl;
}
```

5.2 Simulación

Las simulaciones se han realizado con los alumnos y modelos explicados. Cada simulación consiste en un lote de 10^4 alumnos para minimizar el efecto que tiene el número aleatorio en la simulación. Las simulaciones se han realizado en etapa de autoevaluación, evaluando qué modelo es el más indicado para aprender el temario. Los exámenes de etapa de autoevaluación consisten en un lote de 1000 preguntas (valor configurable). Las simulaciones en la fase de examen es parecido al anterior. Lo que cambia en la fase de examen es que aquí el estudiante no aprende (o es despreciable) y el número de preguntas del examen es 30. En la simulación se han tomado tres niveles de competencia.

Las simulaciones se han realizado con cuatro tipos de alumnos. Estos cuatro alumnos consisten de tres números que representan la probabilidad de 0 a 1 de acertar una pregunta de un nivel específico. En las simulaciones las preguntas están representadas por un número aleatorio. Si este número es menor que el número del alumno para el mismo nivel, se considerará que el alumno ha acertado, mientras que si es mayor, se considerará que el alumno ha fallado a la pregunta.

Todas las simulaciones realizadas siguen el mismo esqueleto. Lo único que cambia en los distintos modelos es la manera en la que se elige el nivel de la siguiente pregunta a presentar al examinado. A continuación se muestra un fragmento de código del “esqueleto” utilizado para la simulación de los distintos modelos. En él se muestra una simulación de un estudiante respondiendo a una pregunta del examen.

Primero se accede al conocimiento actual de un estudiante en un nivel concreto (en la simulación, la probabilidad de que el estudiante responda correctamente a una pregunta del nivel). A continuación se genera un número aleatorio y se comprueba si el número es menor que la probabilidad de que el estudiante acierte la pregunta. Si se cumple la condición se interpreta como una pregunta respondida correctamente, mientras que si no se cumple, se

interpreta que el examinado no ha respondido correctamente a la pregunta. Se registra la respuesta y finalmente se calcula el nivel siguiente (en el modelo) en base a las respuestas respondidas en todos los niveles. La función `recalculateCurrentLevel(current_level, current_student)` es lo único que cambia de un modelo a otro.

```
function answerQuestion(current_level, current_student) {
    float probability = getCurrentKnowledge (current_level,
current_student);
    boolean answer_is_correct = (RANDOM <= probability);
    recordAnswer(current_level, current_student,
answer_is_correct);
    recalculateCurrentLevel(current_level, current_student);
}
```

5.2.1 Configuración

Tanto la simulación como los modelos tienen parámetros configurables. Estos parámetros deberán incluirse en el fichero `simulación.xml`.

```
<?xml version="1.0" encoding="Unicode" standalone="yes"?>
<simulationBean correctAnswersNeeded="70" nrquestions="1000"
times="10000" outputfile="sim-out">
    <model modelClass="PromoteDemote">
        <Parameter name="promote" value="..."/>
        <Parameter name="demote" value="..."/>
    </model>
    <model modelClass="TraditionalLM"/>
    <model modelClass="RewardDirectStudents">
        <Parameter name="param" value="..."/>
    </model>
    <model modelClass="MinimumGrade">
        <Parameter name="param" value="..."/>
    </model>
    ...
    <student name="Direct">
        <initialKnowledge>0.95</initialKnowledge>
        <initialKnowledge>0.7</initialKnowledge>
        <initialKnowledge>0.45</initialKnowledge>
    </student>
    <student ...>
    </student>
</simulationBean>
```

Este fichero contiene los parámetros configurables de la simulación, los modelos a ejecutar con sus respectivos parámetros y los estudiantes a simular con el conocimiento inicial en cada uno de los niveles. Para ver el fichero de configuración utilizado en las simulaciones ir al anexo A.

Parámetros de la simulación:

Parámetro	Descripción
nrquestions	Define el número de preguntas de un examen.
times	Define el número de veces a ejecutar la simulación para cada uno de los modelos para cada alumno.
outputfile	Define el nombre del fichero Excel con los datos resultantes de la simulación.
correctAnswersNeeded	Define el número de respuestas correctas son necesarias para aprobar el examen. Solo usado en fase de evaluación.

Tabla 5-1. Parámetros de la simulación incluidos en el fichero simulación.xml

Parámetros de los modelos:

Parámetro	Descripción
param	Define el parámetro utilizado en el modelo directo y en el modelo de nota mínima.
promote	Define el parámetro de promoción del modelo promoción-democión.
demote	Define el parámetro de democión del modelo promoción-democión.

Tabla 5-2. Parámetros de los modelos incluidos en el fichero simulación.xml

Parámetros de los alumnos:

Parámetro	Descripción
initialKnowledge	Define el conocimiento inicial del alumno en un nivel. Cada alumno contendrá tres parámetros initialKnowledge, al tener tres niveles en la simulación.

Tabla 5-3. Parámetros de los alumnos incluidos en el fichero simulación.xml

5.2.2 Fase de autoevaluación

La simulación realizada trata el grado de aprendizaje de un estudiante después de realizar la prueba de autoevaluación con cada uno de los modelos presentados. Para que la simulación sea representativa de la realidad, se ha tomado un tipo específico de curva de aprendizaje sigmoidea. Esto es porque muchos procesos naturales y curvas de aprendizaje muestran una progresión temporal desde unos niveles bajos al inicio, hasta acercarse a un clímax transcurrido un cierto tiempo; la transición se produce en una región caracterizada por una fuerte aceleración intermedia. La función sigmoidea permite describir esta evolución. [9] En términos de la competencia del estudiante se podría dividir en tres etapas; el proceso de aprendizaje comienza lentamente, aumentando rápidamente hasta estabilizarse al final.

S-Curve (Sigmoid)



Figura 5-3. Curva genérica de la función sigmoidea

En la figura 5-3 se muestra una representación genérica de una curva sigmoidea. El eje horizontal representa la experiencia, ya sea directamente como el tiempo (tiempo de reloj, o el tiempo dedicado a la actividad), o puede estar relacionado con el tiempo (un número de ensayos, o el número total de unidades producidas). En el caso de las simulaciones la experiencia estará expresada como número de preguntas respondidas. El eje vertical representa el aprendizaje o competencia del estudiante. Esto es, un indicador de "eficiencia" o "productividad". En el caso de las simulaciones el aprendizaje será un número de 0 a 1 que indica la competencia del estudiante.

Como se puede observar en la figura 5-3, la primera etapa de aprendizaje del estudiante es lenta y no progresará mucho en la materia. En la segunda etapa el estudiante aprende con rapidez el temario, llegando a un *plateau* en la tercera etapa.

Para realizar la simulación se ha tomado una curva específica de la función sigmoidea: la integral gaussiana. La fórmula de la integral gaussiana se muestra a continuación.

$$\int e^{-x^2} dx$$

A pesar de que no existe ninguna función elemental para dicha fórmula, la integral Gaussiana puede ser resuelta analíticamente con las herramientas del cálculo, por lo que se le ha dado 80 distintos valores en forma de tabla para la simulación.

5.2.3 Fase de evaluación

El código utilizado para realizar las simulaciones es el mismo en la fase de autoevaluación y fase de evaluación. El único cambio entre las dos fases es el número de preguntas. En la etapa de autoevaluación se simulan exámenes con 1000 preguntas, mientras que en la etapa

de evaluación los exámenes simulados contienen tan solo 30 preguntas. Este número de preguntas es configurable.

Debido a que el número de preguntas en etapa de evaluación es tan bajo, el aprendizaje después de este tipo de exámenes es despreciable y no se tiene en cuenta.

6 Integración, pruebas y resultados

6.1 Parámetros utilizados

A continuación se muestran los parámetros utilizados en la simulación para cada uno de los modelos. Estos parámetros se han utilizado tanto para las simulaciones en etapa de autoevaluación como en la etapa de evaluación.

6.1.1 Modelo tradicional

No tiene parámetros.

6.1.2 Modelo promoción-democión

En el modelo promoción-democión hay dos variables que definen el número de preguntas que hay que responder correctamente para subir de nivel y el número de preguntas que hay que responder incorrectamente para bajar de nivel. Los valores dados a estas variables en la simulación se muestran en la tabla 6-1.

Nombre del algoritmo	Número de respuestas correctas para subir de nivel	Número de respuestas incorrectas para bajar de nivel
PD-1-1	1	1
PD-1-2	1	2
PD-1-3	1	3
PD-2-1	2	1
PD-3-1	3	1

Tabla 6-1. Definición del algoritmo de promoción-democión con los parámetros usados

6.1.3 Modelo directo

El modelo directo consta de un solo parámetro que regula el conocimiento que el estudiante tiene que obtener en el nivel actual para pasar al nivel siguiente, en función del conocimiento que tiene en el nivel siguiente. En la tabla 6-2 se muestra los distintos valores que se le ha dado al parámetro p en la simulación.

Nombre del algoritmo	Valor del parámetro
CD-0.5	0.5
CD-1	1
CD-1.5	1.5
CD-2	2
CD-2.5	2.5
CD-3	3
CD-4	4
CD-6	6
CD-8	8

Tabla 6-2. Definición del algoritmo directo con su parámetro p

6.1.4 Modelo nota mínima

El modelo de nota mínima consta de un solo parámetro que regula el conocimiento mínimo que el estudiante tiene que obtener en el nivel actual para pasar al nivel siguiente.

Nombre del algoritmo	Valor del parámetro
MG-0.6	0.6
MG-0.7	0.7
MG-0.8	0.8

Tabla 6-3. Definición del algoritmo de nota mínima con su parámetro p

6.2 Fase de autoevaluación

Las gráficas resultantes son generadas automáticamente importando el archivo de salida (parámetro outputfile del fichero de configuración) en la hoja de cálculo LearnEval.xlsx. Muestran los resultados obtenidos de las simulaciones del aprendizaje de cada tipo de alumno después de realizar el examen de autoevaluación con cada uno de los modelos y los parámetros expuestos.

Los datos que se han considerado importantes a la hora de mostrar los resultados son:

- El aprendizaje de los tipos de estudiantes en cada nivel de conocimiento después de realizar una prueba de autoevaluación.
- La media del aprendizaje de los tipos de estudiantes en todos los niveles.
- *Directness*¹ de los estudiantes después de realizar la prueba de autoevaluación. Resulta útil para diferenciar a los alumnos directos e inversos antes y después de realizar la prueba.

Primero se muestran las gráficas del aprendizaje de los cuatro tipos de estudiantes por nivel. Debido a que el conocimiento inicial de los estudiantes es fijo (consultar el apartado 4.2 para ver el conocimiento inicial de cada uno de los estudiantes) para todos los modelos, se ha omitido de las gráficas.

Las gráficas representan el incremento en el control de la materia por niveles. La gráfica 6-1 muestra que en nivel 1 sobretodo aprenden los estudiantes inversos y los malos, lo cual es lógico porque son los que más tienen aprender en el nivel 1. Sin embargo se ve claramente que para que estos tipos de estudiantes aprendan en el nivel 1 habría que evitar los modelos PD-1-1, PD-1-2 y PD-1-3 ya que estos modelos no fomentan el aprendizaje. Los modelos óptimos parecen CD-6.0 y CD-8.0.

¹ Directness - término utilizado para definir cómo de directo es un estudiante. Se calcula comparando el nivel de habilidad en un nivel con el del nivel siguiente. El estudiante directo tiene un directness de 1, mientras que el estudiante indirecto tiene un directness de 0.

La gráfica 6-2 no muestra una clara diferencia entre modelos. En esta gráfica se presentan los datos del aprendizaje en el nivel 2. Se puede observar que con los modelos de nota mínima MG-0.6 y MG-0.7 los estudiantes no aprenden. Esto es porque en este modelo el aprendizaje del estudiante se centra en el nivel 1, por lo que no tiene mucho que aprender en ese nivel y, por tanto, se “sacrifica” el resto de niveles. La gráfica 6-3 es casi el complemento matemático de la gráfica 6-1. Esto tiene su lógica, ya que los estudiantes siempre aprenden, y el modelo determina en qué nivel aprenden. Si no aprenden en el primer nivel, aprenden en el tercero. La gráfica 6-4 representa el aprendizaje de los estudiantes en todos los niveles.

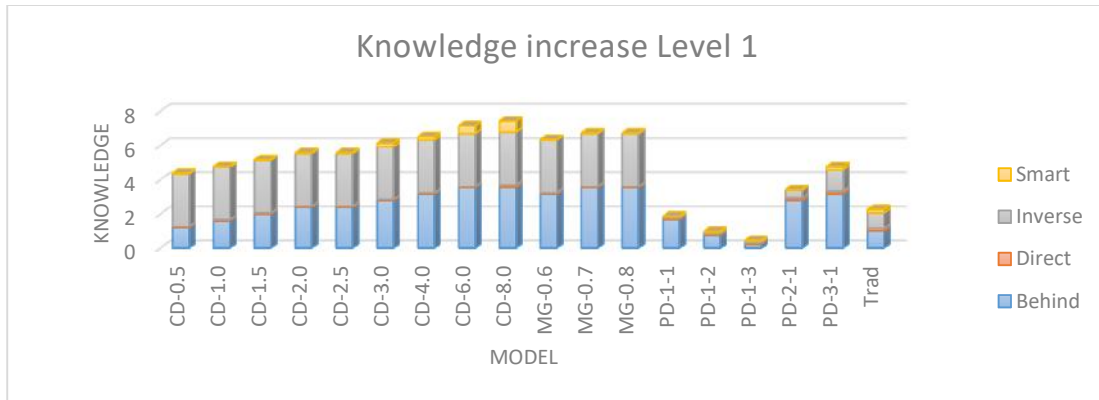


Figura 6-1. Aprendizaje en el nivel 1 de todos los estudiantes por modelo después de realizar la prueba de autoevaluación

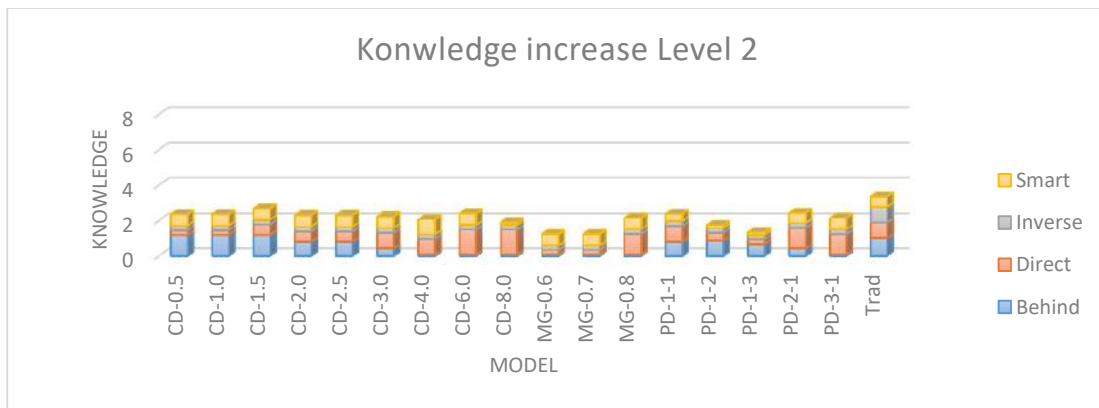


Figura 6-2. Aprendizaje en el nivel 2 de todos los estudiantes por modelo después de realizar la prueba de autoevaluación

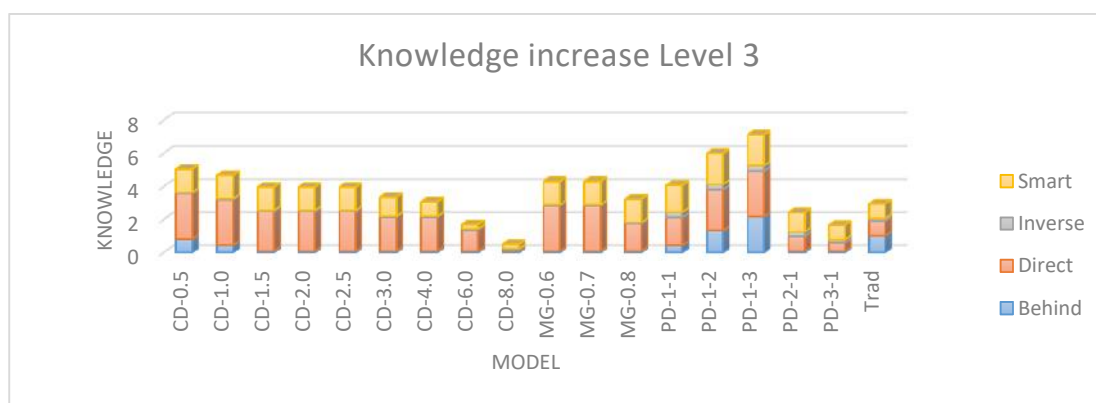


Figura 6-3. Aprendizaje en el nivel 3 de todos los estudiantes por modelo después de realizar la prueba de autoevaluación

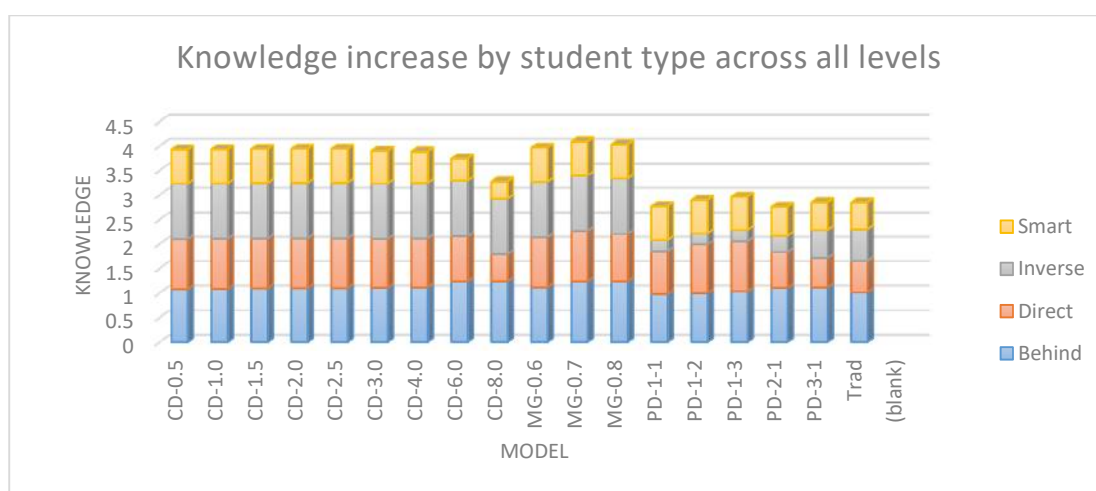


Figura 6-4. Aprendizaje en todos los niveles de los estudiantes por modelo después de realizar la prueba de autoevaluación

A continuación se exponen las gráficas de la comparación del conocimiento inicial con el aprendizaje de cada uno de los estudiantes. Estas gráficas contienen la media del aprendizaje de todos los niveles. Como se puede observar en la gráfica 6-5 los modelos que más benefician al estudiante directo (con conocimiento inicial [0.95, 0.7, 0.45]) son los modelos directos CD-0.5 a CD-6.0 y los modelos de promoción-democión PD-1-2 y PD-1-3.

El modelo directo beneficia al estudiante directo porque se centra los niveles básicos antes de pasar a preguntas de niveles superiores. Esto quiere decir que el alumno directo responde correctamente a la mayoría de las preguntas del nivel básico. En el segundo nivel también responde correctamente a un gran número de preguntas, lo que le permite llegar al tercer nivel. Dado que el conocimiento del alumno directo en el nivel avanzado es el más bajo de los tres, es aquí donde tiene mucho que aprender y donde aumenta considerablemente su conocimiento.

El modelo de promoción-democión beneficia considerablemente al estudiante directo en los casos PD-1-2 y PD-1-3. En estos casos el número de preguntas correctas que tiene que obtener para subir de nivel es 1, mientras que el número de preguntas incorrectas que tiene que obtener para bajar de nivel es 2 y 3 respectivamente. Este modelo beneficia al estudiante directo porque éste tiene mucho que aprender en los niveles avanzados y no tanto en el nivel básico. Tener $nPromocion \leq nDemocion$ implica que al estudiante le cueste más bajar de nivel que subir de nivel. Esto es, que pase más tiempo en los niveles más altos (si es que llega) que en los bajos y, por tanto, aprenda más en estos niveles, que es lo que necesita el estudiante directo. Los modelos de nota mínima benefician al estudiante directo ya que el único nivel en los que no ha alcanzado la nota mínima es en el tercer nivel así que pasará más tiempo en este nivel, aprendiendo.

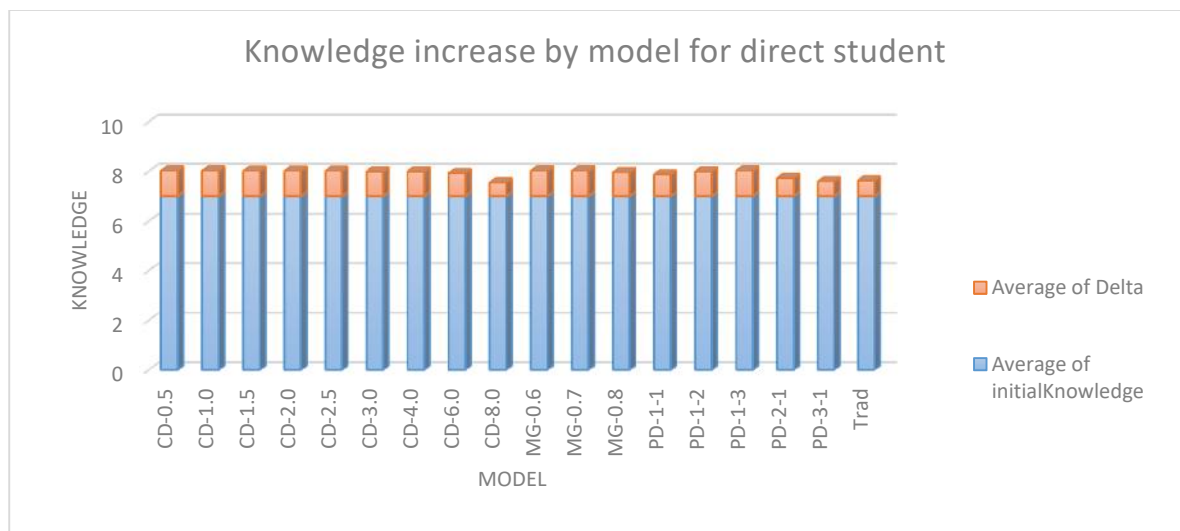


Figura 6-5. Aprendizaje del estudiante directo por modelo después de realizar la prueba de autoevaluación

En la gráfica 6-6 se muestran los resultados del estudiante inverso. Al estudiante inverso (con conocimiento inicial [0.45, 0.7, 0.95]) le beneficia el modelo directo, ya que con este modelo responde principalmente preguntas del nivel básico. Debido a que en el nivel básico le queda bastante por aprender, es necesario que pase la mayoría del tiempo (si es que no todo) en los niveles inferiores.

El modelo de promoción-democión con los parámetros PD-1-1, PD-1-2 y PD-1-3 hace lo contrario de lo que necesita el estudiante, ya que se centra en los niveles más altos. Y debido a que ya tiene un alto conocimiento en estos niveles, no tiene mucho que aprender en los mismos. Sin embargo, se puede observar que PD-2-1 y PD-3-1 sí beneficia al estudiante inverso, aunque en menor medida. Esto es porque aunque sí se pase mucho tiempo en el nivel básico, para llegar a él tiene que fallar una pregunta de cada nivel superior (lo cual es menos probable, ya que su conocimiento es bastante alto en los niveles avanzados). Los modelos MG-0.6 y MG-0.7 benefician al estudiante inverso por la misma razón.

Comparando las dos gráficas de estudiantes directos e inversos se puede observar que el aprendizaje con el modelo tradicional es el mismo, ya que en este modelo no se diferencia entre estos dos tipos de estudiantes.

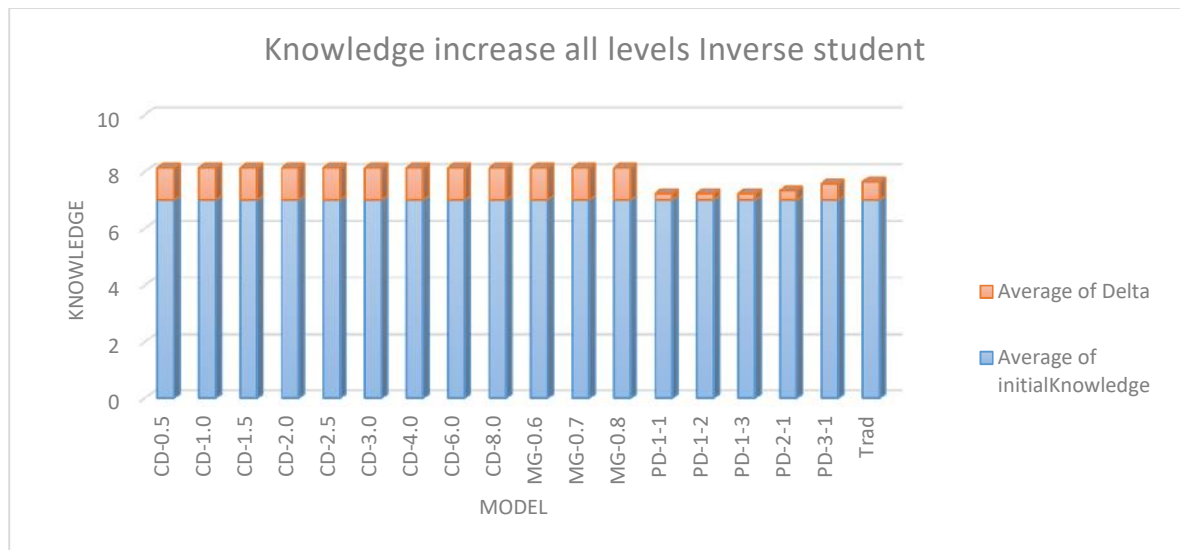


Figura 6-6. Aprendizaje del estudiante inverso por modelo después de realizar la prueba de autoevaluación

El estudiante bueno, mostrado en la gráfica 6-7, no tiene mucho que aprender, ya que su conocimiento es bastante alto en los tres niveles ([0.9, 0.8, 0.7]). Aun así hay un poco de diferencia en el aprendizaje en los distintos modelos. Los modelos que más benefician al estudiante bueno son promoción-democión PD-1-1, PD-1-2 y PD-1-3. Esto es por la misma razón que beneficia al estudiante directo. Tanto el estudiante directo como el estudiante bueno tienen más que aprender en el nivel avanzado. Los modelos directos CD-0.5 a CD-4.0 también benefician al estudiante bueno debido a que se centra en aprender en los niveles avanzados, en vez de centrarse en los niveles inferiores, en los cuales tiene mayor conocimiento. El modelo de nota mínima también beneficiará al estudiante bueno.

El estudiante malo (con conocimiento en los niveles [0.3, 0.3, 0.3]), mostrado en la gráfica 6-8, es el que más tiene que aprender. Debido a que tiene el mismo nivel de conocimiento en todos los niveles, se beneficiará considerablemente de todos los modelos. Los modelos que más le benefician son los modelos directos CD-6.0 y CD-8.0. Esto es porque cuanto más alto sea el parámetro, más se tiene que acercar a 1 el conocimiento en el nivel básico. La ventaja del modelo directo para el estudiante malo es que la mayoría del aprendizaje ocurre en el nivel básico creando, en mayor o menor medida, estudiantes directos.

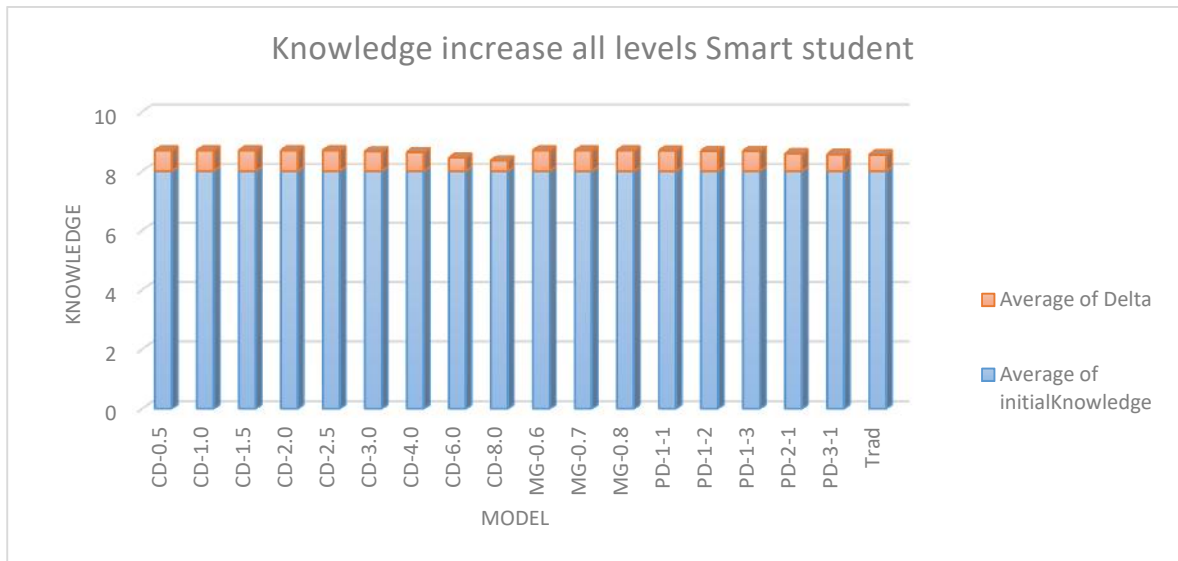


Figura 6-7. Aprendizaje del estudiante bueno por modelo después de realizar la prueba de autoevaluación

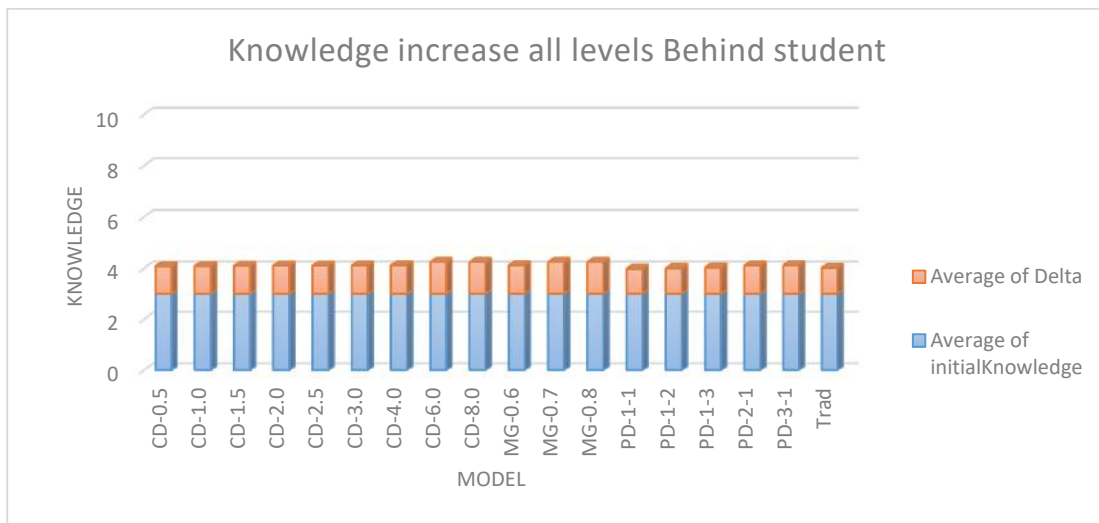


Figura 6-8. Aprendizaje del estudiante malo por modelo después de realizar la prueba de autoevaluación

Se puede concluir, entonces, que el modelo óptimo es CD-6.0. Este algoritmo promueve el aprendizaje de la materia básica antes de la avanzada y si el estudiante ya tiene un conocimiento de la materia básica pasa a la avanzada, aprendiendo el temario en el orden ideal. Así prevenimos que los alumnos malos se conviertan en alumnos inversos y se intenta crear, a medida de lo posible, alumnos directos.

6.3 Fase de evaluación

En la fase de evaluación el número de preguntas es inferior que en autoevaluación, por lo que el efecto de aprendizaje es despreciable. La grafica 6-9 se genera automáticamente importando el archivo de salida de la simulación en la hoja Excel TestEval.xlsx.

En la gráfica 6-9 se simulan los mismos modelos que en la etapa de autoevaluación. Esta gráfica representa el porcentaje de estudiantes que aprueba el examen. Como simulación de control se ha incluido el modelo tradicional. La simulación se ha ejecutado con 10000 estudiantes y un examen de 30 preguntas de opción múltiple. Se ha configurado que para aprobar hace falta contestar correctamente al menos el 70% de las preguntas.

En [8] se pretende beneficiar a los estudiantes directos sobre los inversos, ya que se considera más importante conocer el temario básico que el avanzado. En la gráfica 6-9 se puede ver que en el modelo tradicional aprueban el examen el mismo número de estudiantes directos que inversos, que es lo que se intenta evitar. Sin embargo, con los modelos PD-2-1 y PD-3-1 el mismo número de estudiantes buenos y directos aprueban, lo cual tampoco es justo ya que los estudiantes buenos tienen un mayor conocimiento de la materia que los directos. Los modelos PD-1-1, PD-1-2 y PD-1-3 también son malos en fase de evaluación ya que existe un mayor porcentaje de estudiantes inversos que aprueban e incluso éste es mayor que el porcentaje de estudiantes buenos que aprueban.

El estudiante malo no aprueba con ninguno de los modelos, ya que se ha configurado que para aprobar hay que contestar correctamente al menos el 70% de las preguntas. Esto reduce dramáticamente la probabilidad de que el estudiante malo apruebe, ya que conoce tan solo 1/3 de la materia en los tres niveles. Para que el estudiante malo apruebe con algunos modelos, habría que cambiar la configuración y reducir el porcentaje de preguntas que hay que contestar correctamente para aprobar.

Por tanto, si se busca un equilibrio en el que aprueben estudiantes directos y buenos los modelos CD-6.0, CD-8.0 y MG-0.8 son óptimos.

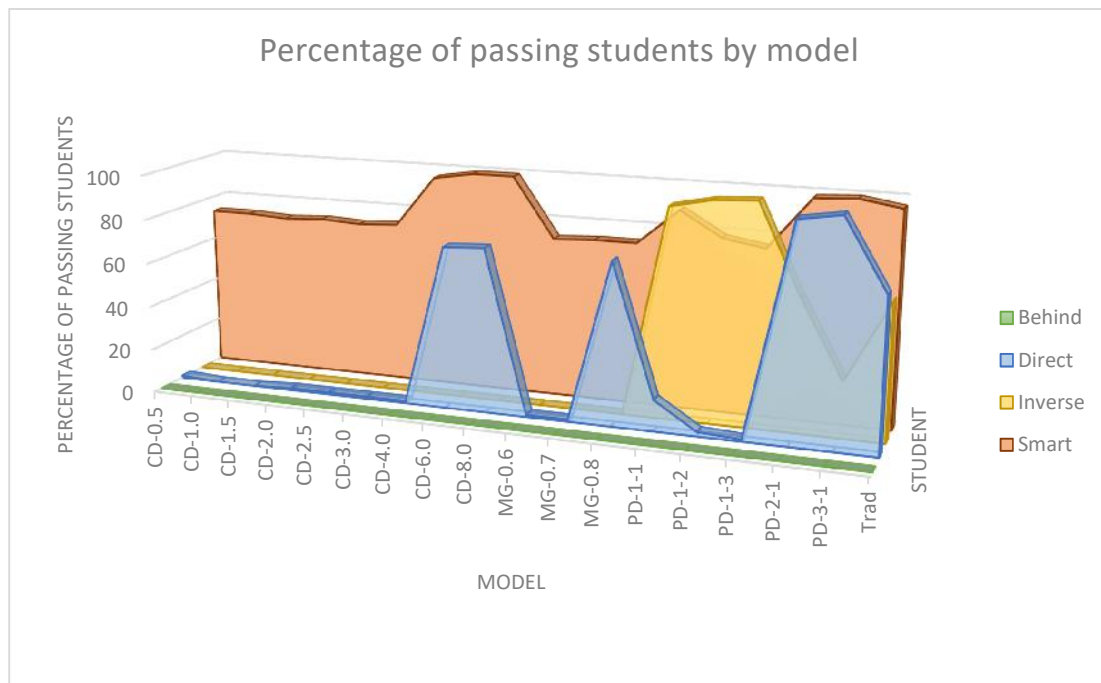


Figura 6-9. Porcentaje de los estudiantes que aprueban en cada modelo

7 Conclusiones y trabajo futuro

7.1 Conclusiones

Se ha trabajado mucho en el área de los exámenes adaptativos CAT. Este proyecto añade un método estructurado en el que se determina la validez de los modelos adaptativos. Tanto la simulación como los modelos son configurables y el sistema es lo suficientemente genérico para evaluar cualquier modelo.

Algunos de los modelos implementados tienen resultados prometedores. El modelo directo con parámetro 6.0 parece ser óptimo para la etapa de evaluación como de autoevaluación. Este modelo promueve el aprendizaje de la materia básica antes de la avanzada, aprendiendo el temario en el orden ideal. Así se evita que los alumnos malos se conviertan en alumnos inversos y se intenta crear, a medida de lo posible, alumnos directos. En la etapa de evaluación el modelo proporciona un equilibrio en el que aprueban estudiantes directos y buenos.

Una observación importante es que casi todos los modelos incluidos en estas simulaciones presentan resultados mejores que el modelo tradicional. En la mayoría de modelos el aprendizaje del estudiante está focalizado a sus necesidades, mientras que con el modelo tradicional se presenta al estudiante con 1/3 de preguntas de cada nivel, sin tener en cuenta el conocimiento previo del alumno. En la etapa de evaluación se puede observar que los alumnos directos e inversos responden correctamente al mismo número de preguntas, que es lo que se intenta evitar en [8].

7.2 Trabajo futuro

Integración en una aplicación AEH

El siguiente paso de esta aplicación y los modelos propuestos es incluirlos en una aplicación real como, por ejemplo, e-valUAM. e-valUAM es un sistema adaptativo orientado a crear cuestionarios que sirvan de apoyo al aprendizaje, ya sea como sistema de autoevaluación de los alumnos o como sistema de evaluación para los docentes. [1] Utilizando una herramienta de este estilo se podrían realizar pruebas con usuarios reales. Además, habría más opciones de modelos de exámenes tipo test.

La integración de los modelos en una aplicación real es relativamente sencilla por parte de los modelos aquí implementados. Para integrar el modelo promoción-democión no es necesario conocer nada del estudiante, por lo que simplemente habría que coger las preguntas divididas en niveles y aplicar el modelo. Sin embargo, para el modelo directo y el modelo de nota mínima se tiene que conocer el conocimiento inicial del estudiante, por lo que para aplicar este modelo habrá que hacer una breve prueba inicial donde se estime el conocimiento inicial del alumno, el cual es usado a la hora de aplicar el modelo.

Referencias

- [1] Pablo Molins Ruano “Desarrollo de un sistema de cuestionarios adaptativos para el apoyo al aprendizaje” Trabajo de Fin de Grado, Universidad Autónoma de Madrid, pp. 1-2, 7-11, 25, 2013.
- [2] David J. Weiss, G. Gage Kingsbury, “Application of Computerized Adaptive Testing to Educational Problems” *Journal of Educational Measurement* 21(4):361 – 375, September 2005.
- [3] H. Wainer, N. J. Dorans, D. Eignor, R. Flaugher, B. F. Green, R. J. Mislevy, L. Steinberg, and D. Thissen, “Computer-Adaptive Testing: A Primer”, Routledge, Ed. Mahwah, NJ, USA: Lawrence Erlbaum Associates, pp. 1-17, 37-46, 61-68, 101-106, 2000.
- [4] F. M. Lord, “Some test theory for tailored testing” *ETS Research Bulletin Series*, vol. 1968, no. 2, pp. 60-62, 1968.
- [5] Larson, J. W, “Computer –assisted language testing: is it portable?” *ADFL Bulletin*, 18(2), pp. 20-24, 1987.
- [6] Margit Antal, Levente Er s, Attila Imre “Computerized adaptive testing: implementation issues” *Acta Univ. Sapientiae Informatica*, vol. 2 no. 2, pp. 168-183, 2010.
- [7] H. Wainer, “On item response theory and computerized adaptive tests.” *Journal of College Admissions*, vol. 27, no. 4, pp. 9-16, 1983.
- [8] C. González-Sacristán, P. Molins-Ruano, F. Díez, P. Rodríguez and G. M. Sacha, “Computer-aided assessments for basic programming subjects”, Departamento de Ingeniería Informática. Escuela Politécnica Superior. Universidad Autónoma de Madrid, 2013.
- [9] Escolano, Francisco, “Inteligencia artificial”. Editorial Paraninfo, p. 96, 2003.
- [10] Frank B. Baker, “The Basics of Item Response Theory”, Second edition Ed. the ERIC Clearinghouse on Assessment and Evaluation, pp. 130-132, 2001.
- [11] S. Klinkenberg*, M. Straatemeier, H.L.J. van der Maas, “Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation” *Science Direct*, pp. 1813-1824, 2011.

Glosario

AEH Autoría de hipermedio adaptativo, por sus siglas en inglés, *Adaptive Educational Hypermedia*.

CAT Test Adaptativos por ordenador, por sus siglas en inglés, *Computerized adaptive testing*.

CBT Aprendizaje por ordenador, por sus siglas en inglés, *Computer based learning*.

IRT Teoría de respuesta al ítem, por sus siglas en inglés, *Item response theory*.

P&P Exámenes tradicionales realizados con papel y lápiz, de sus siglas en inglés, *Conventional paper and pencil tests*.

TEL Aprendizaje asistido por tecnologías, por sus siglas en inglés, *Technology-Enhanced Learning*.

Fuente:

http://www.ncme.org/ncme/NCME/Resource_Center/Glossary/NCME/Resource_Center/Glossary1.aspx?hkey=4bb87415-44dc-4088-9ed9-e8515326a061#anchorA

Anexos

A Manual de instalación

Descargar el archivo simulator.jar y ejecutarlo con el siguiente comando:

```
java -cp simulator.jar simulation.Simulator
```

Para la ejecución del programa es necesario el fichero simulation.xml que indica los parámetros con los que hay que ejecutar cada modelo, el número de preguntas de cada examen, el número de veces que se debe ejecutar y el nombre del fichero CSV de salida. El fichero utilizado para las simulaciones se muestra a continuación.

```
<?xml version="1.0" encoding="Unicode" standalone="yes"?>
<simulationBean correctAnswersNeeded="70" nrquestions="1000" times="10000"
outputfile="sim-out">
  <model modelClass="PromoteDemote">
    <Parameter name="promote" value="1"/>
    <Parameter name="demote" value="1"/>
  </model>
  <model modelClass="PromoteDemote">
    <Parameter name="promote" value="1"/>
    <Parameter name="demote" value="2"/>
  </model>
  <model modelClass="PromoteDemote">
    <Parameter name="promote" value="1"/>
    <Parameter name="demote" value="3"/>
  </model>
  <model modelClass="PromoteDemote">
    <Parameter name="promote" value="2"/>
    <Parameter name="demote" value="1"/>
  </model>
  <model modelClass="PromoteDemote">
    <Parameter name="promote" value="3"/>
    <Parameter name="demote" value="1"/>
  </model>
  <model modelClass="TraditionalLM"/>
  <model modelClass="RewardDirectStudents">
    <Parameter name="param" value="0.5"/>
  </model>
  <model modelClass="RewardDirectStudents">
    <Parameter name="param" value="1"/>
  </model>
  <model modelClass="RewardDirectStudents">
    <Parameter name="param" value="1.5"/>
  </model>
  <model modelClass="RewardDirectStudents">
    <Parameter name="param" value="2"/>
  </model>
  <model modelClass="RewardDirectStudents">
    <Parameter name="param" value="2.5"/>
  </model>
</simulationBean>
```

```

</model>
<model modelClass="RewardDirectStudents">
  <Parameter name="param" value="3"/>
</model>
<model modelClass="RewardDirectStudents">
  <Parameter name="param" value="4"/>
</model>
<model modelClass="RewardDirectStudents">
  <Parameter name="param" value="6"/>
</model>
<model modelClass="RewardDirectStudents">
  <Parameter name="param" value="8"/>
</model>
<model modelClass="RewardDirectStudents">
  <Parameter name="param" value="10"/>
</model>
<model modelClass="MinimumGrade">
  <Parameter name="param" value="0.6"/>
</model>
<model modelClass="MinimumGrade">
  <Parameter name="param" value="0.7"/>
</model>
<model modelClass="MinimumGrade">
  <Parameter name="param" value="0.8"/>
</model>
<student name="Direct">
  <initialKnowledge>0.95</initialKnowledge>
  <initialKnowledge>0.7</initialKnowledge>
  <initialKnowledge>0.45</initialKnowledge>
</student>
<student name="Inverse">
  <initialKnowledge>0.45</initialKnowledge>
  <initialKnowledge>0.7</initialKnowledge>
  <initialKnowledge>0.95</initialKnowledge>
</student>
<student name="Smart">
  <initialKnowledge>0.9</initialKnowledge>
  <initialKnowledge>0.8</initialKnowledge>
  <initialKnowledge>0.7</initialKnowledge>
</student>
<student name="Behind">
  <initialKnowledge>0.3</initialKnowledge>
  <initialKnowledge>0.3</initialKnowledge>
  <initialKnowledge>0.3</initialKnowledge>
</student>
</simulationBean>

```

El fichero de salida que contiene los datos resultantes de la simulación tiene formato CSV y se puede visualizar con una herramienta como Excel. Para visualizar las gráficas mostradas en los resultados, es posible importar estos datos en el fichero TestEval.xlsx y LearnEval.xlsx. Estas hojas de cálculo generan las gráficas explicadas en los apartados 6.2 y 6.3.

B Manual del programador

A continuación se describen los pasos a tomar para la integración de modelos externos en la aplicación.

La integración de modelos externos es relativamente sencilla, pero tiene algunas limitaciones. Lo que se ha tenido en cuenta a la hora de elegir el siguiente ítem a mostrar al examinado es únicamente el nivel de dicho ítem. La limitación del modelo viene dada por lo único que decide el modelo (y la parte más extensa de la simulación) es el nivel de la siguiente pregunta a mostrar.

Para la integración de más modelos se deberá crear una clase que herede de `TestModel` e implemente el método `int recalculateCurrentLevel (int curlevel, List<Level> levels);` El modelo deberá definirse en el fichero XML y si tiene parámetros, estos también tendrán que estar definidos en el mismo.

```
public interface TestModel {  
    int recalculateCurrentLevel (int curlevel, List<Level>  
levels);  
    public String getParams();  
    void init(ModelBean mb);  
}
```